4.1分类—两个步骤

- 模型的建立:
  - 给定一个事先确定类别属性的数据集合
  - 每一个实体假设属于一个事先定义好的类别，也就是说它们的类别属性的值是确定的
  - 用来构造模型的实体集合我们把它叫做 训练集合
  - 模型以分类规则, 决策树,或者是数学函数的形式给出

- 模型的使用: 用来对未来或者是未知的数据进行分类
  - 模型精度的评估
    - 用分类结果与测试集合实际的类别属性值进行比较
    - 分类精度的值为测试集合中被分类模型正确分类的实体在测试集合中所占的比率
    - 测试集合与分类集合相互独立, 否则将会发生过拟合
  - 如果精度的值是可以接受的, 那么将使用此分类模型对那些决策属性未知的实体进行分类

4.1监督学习与非监督学习

- 监督学习 (分类)
  - 训练数据集合 (观察，测量等方法得到) 的类别属性是由观察得到的，是已知的。
  - 新的数据集合由基于训练集合得到的分类模型进行分类。
- 非监督学习 (聚类)
  - 训练集合的类别属性是未知的。
  - 给出一个由观察或测量等方法得到的数据集合。在数据中建立已经存在的类或簇。

4.2 与分类和预测相关的几个问题

- **（1） 数据预处理**
- 数据清理
  - 数据预处理的目的是减少噪音和处理缺失数据
- 数据分析 (特征选择)
  - 删除无关的、冗余的属性，选择相关属性
- 数据变换
  - 泛化
  - 规范化数据

4.2 与分类和预测相关的几个问题

（2）分类方法的评估

- 预测精度
- 时间性能
  - 建立模型所花费时间
  - 使用模型所花费时间
- 健壮性
  - 处理噪音和缺失数据的能力
- 空间性能
  - 处理驻磁盘数据库的能力等
- 可理解性:
  - 模型给出的结果的可理解性
- 规则的评估
  - 模型规模大小
  - 分类规则的简洁性

4.2 与分类和预测相关的几个问题

（2）分类方法的评估

- 预测精度
- 实际中正例的个数为TP+FN

被当作+的+样本	
TP ++	TN --
FP +-	FN -+

1）正确率（accuracy）

accuracy = （TP+TN）/(P+N)

2）错误率（error rate)

error rate = (FP+FN)/(P+N)

accuracy =1 - error rate

3）灵敏度（sensitive）

sensitive = TP/P，表示的是所有正例中被分对的比例，衡量了分类器对正例的识别能力

4.2 与分类和预测相关的几个问题

（2）分类方法的评估

- 预测精度
- 实际中正例的个数为TP+FN

4）特效度（specificity)

specificity = TN/N，表示的是所有负例中被分对的比例，衡量了分类器对负例的识别能力；

5）精度（precision）

精度是精确性的度量，表示被分为正例的样本中实际为正例的比例，precision=TP/（TP+FP）；

6）召回率（recall）

召回率是覆盖面的度量，度量有多个正例被分为正例，recall=TP/(TP+FN)=TP/P=sensitive，可以看到召回率与灵敏度是一样的。

4.2 与分类和预测相关的几个问题

（2）分类方法的评估

- 预测精度
- 实际中正例的个数为TP+FN

TP	TN
FP	FN

指标难以同时都越大越好，因此需要综合指标来评估

7）F1-score

是模型精确率和召回率的加权平均

F1=2*precision*recall/(precision+recall)

4.2 与分类和预测相关的几个问题

（2）分类方法的评估

- 预测精度
- 实际中正例的个数为TP+FN

TP	TN
FP	FN

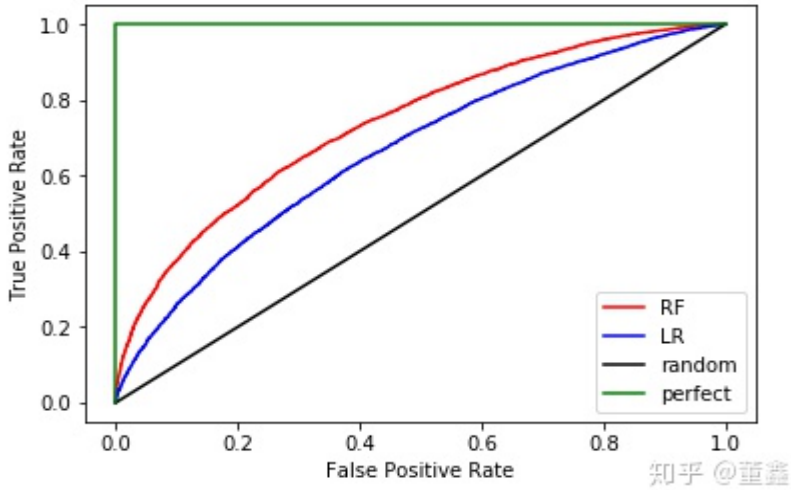
8）ROC(Receiver operating Characteristic Curve)

横坐标表示假阳性率**FPR**，纵坐标表示正阳性率**TPR**。

TPR=TP/(TP+FN)

FPR=FP/(FP+TN)

网络访问中的入侵检测评估，采用哪个指标比较合适 ROC



ROC同时关注正负样本,在imbalance 数据集上更敏感。

ROC曲线越靠拢(0,1)点，越偏离45度对角线越好

4.2 分类评估指标中的代价——cost

垃圾邮件检测

正例检测为负例

负例检测为正例

癌症疾病诊断中

正例检测为负例

负例检测为正例

分类与预测——常用的分类与预测算法

- 主要分类与预测算法简介：

算法名称	算法描述
回归分析	回归分析是确定预测属性（数值型）与其他变量间相互依赖的定量。关系的最常用的统计学方法。包括线性回归、非线性回归、Logistic回归、岭回归、主成分回归、偏最小二乘回归等模型。
决策树	它采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较，并根据不同的属性值从该结点向下分支，叶结点是要学习划分的类。
人工神经网络	一种模仿大脑神经网络结构和功能而建立的信息处理系统，表示神经网络的输入与输出变量之间关系的模型。
贝叶斯网络	贝叶斯网络又称信度网络，是Bayes方法的扩展，是目前不确定知识表达和推理领域最有效的理论模型之一。
支持向量机	SVM支持向量机根据有限的样本信息在模型的复杂性和学习能力之间寻求最佳折衷，以获得最好的推广能力。

4.3 决策树分类

4.3.1 决策树构造算法CLS

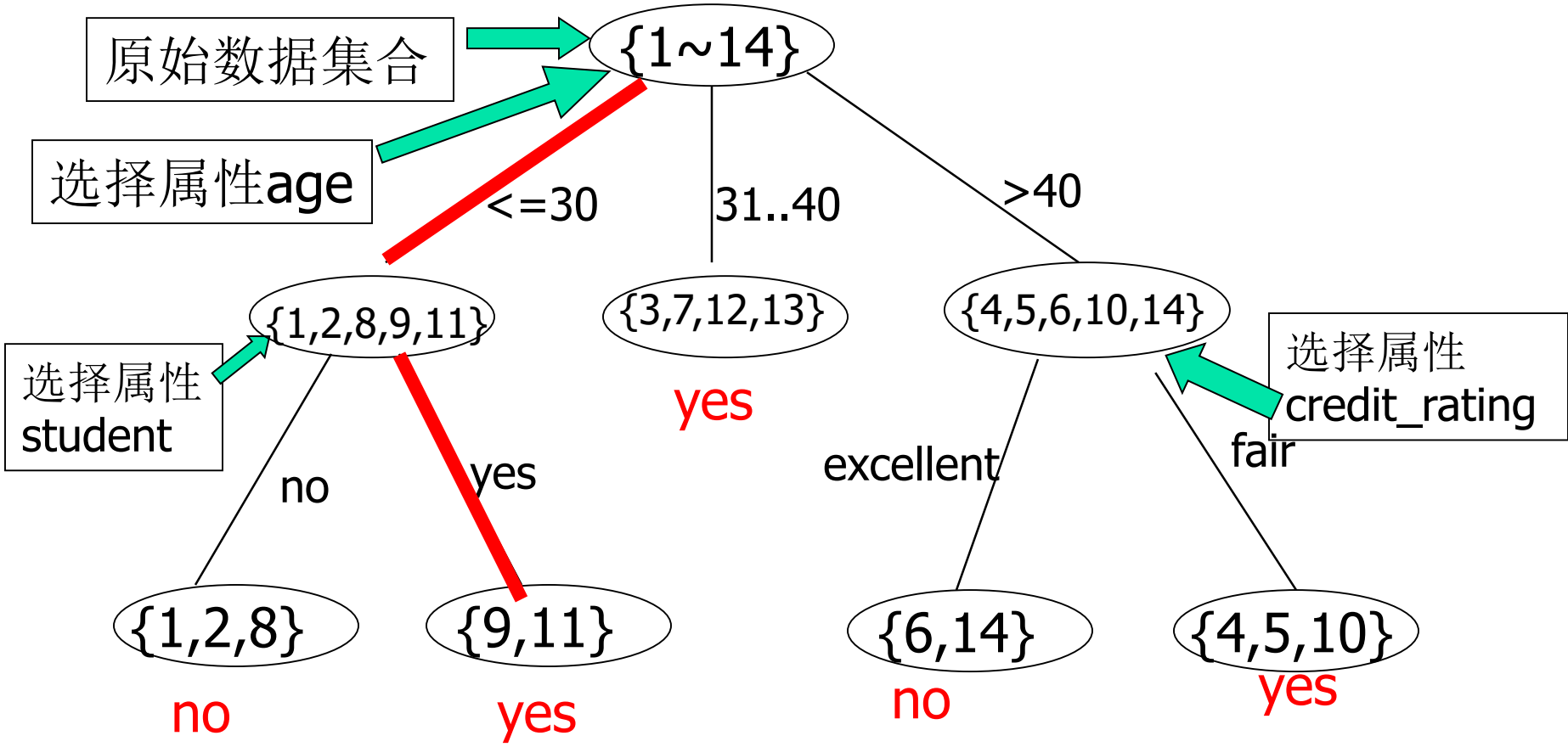
- 决策树（Decision Tree）是一种最常用的分类知识描述模型
- 决策树是一棵树结构:
  - 每个分支结点对应一个对象集合和一个属性选择，
  - 每个分支对应一个测试条件，
  - 叶子结点对应一个类别判断。

- Hunt在66年提出的概念学习系统（Concept Learning System, CLS）中，给出了基于决策树的学习。
- 该算法接受一组训练实例，构造出决策树。
- 决策树从根结点到叶子结点可得到有关概念的一个描述，将类型相同的各个叶子结点所对应的描述合在一起可构成该概念的所有描述。
- 构造过程采用的是自上而下的，递归的，分而治之的方法。

4.3.1 决策树构造算法CLS

- CLS决策树算法的主要步骤如下：
  - 1）T:=所有训练实例集合，产生一个结点T；
  - 2）如果T中所有实例都是正例，
    则为该结点产生一个“yes”标志，并结束；
  - 3）如果T中所有实例都是反例，
    则产生一个“no”标志，并结束；
  - 4）否则，选择一个属性X，
    - 假设其所有取值为$v_1,v_2,\ldots,v_n$，
    - 则依据这些取值，将T划分为n个子集$T_1,T_2,\ldots,T_n$，
    - 建T的n个孩子结点T_i，并分别以$X= v_i$作为从T到T_i的分支标号。
  - 5）对每个T_i，以递归的方式建一棵以T_i为根的子树。

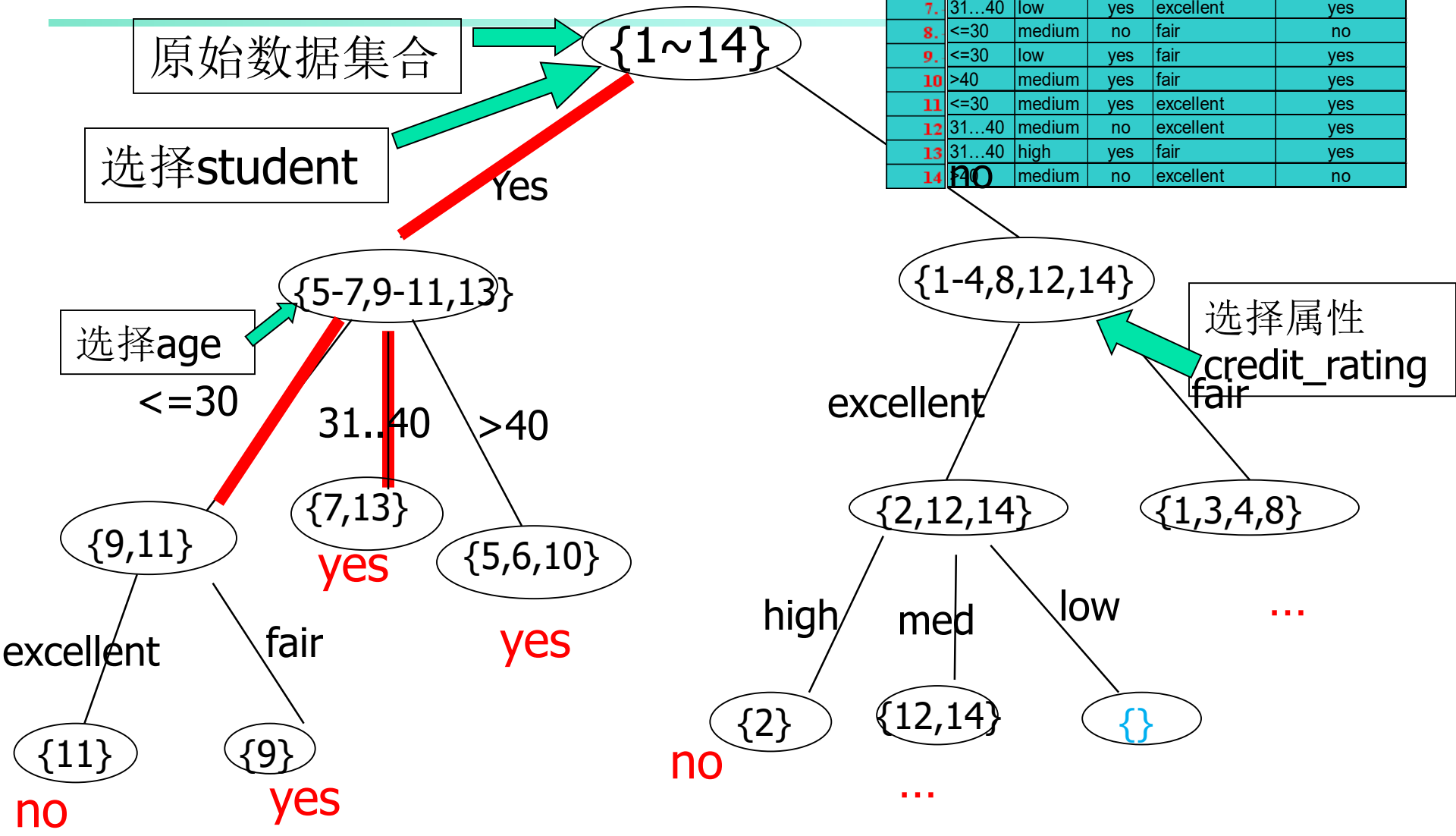
关于是否购买电脑的决策树的过程与结果



分类规则：if age<=30 and student=yes then buys_computer=yes；

关于是否购买电脑的决策树的过程与结果

NO.	age	income	student	credit_rating	buys_computer
1.	<=30	high	no	fair	no
2.	<=30	high	no	excellent	no
3.	31…40	high	no	fair	yes
4.	>40	medium	no	fair	yes
5.	>40	low	yes	fair	yes
6.	>40	low	yes	excellent	no
7.	31…40	low	yes	excellent	yes
8.	<=30	medium	no	fair	no
9.	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no



4.3.1 决策树构造算法CLS

CLS算法分析

- 显然，这不是唯一的解，还可以有其它形式的决策树。
- 由实例可以看出，这一算法较为简捷、直观，但也存在**问题**：
  - 抗干扰能力弱：
    - 噪音数据难免，算法对噪音敏感
  - 易受无关属性的影响，导致规则烦琐。
  - 受属性选择次序的影响：
    - 各条件属性对分类的影响程度可能存在差异。

      选择次序得当，将使规则能反映其关系的实质，

      但如果将影响因素小的在前面展开，将导致规则条件烦琐，不能反映其内在联系。
  - 只能发现部分规则。

4.3.2 决策树构造算法ID3

- 学者Quinlan 于1986年提出了构造决策树的ID3算法，采用启发式函数计算出属性选择次序。
- ID3算法的求解步骤与CLS类似，描述如下：
  - 1）T:=所有训练实例集合，产生一个结点T；
  - 2）如果T中所有实例都属于一个类别，则为该结点产生一个类别标记，并结束；
  - 3）否则，选择**具有最高信息增益的**属性X，
    - 假设其所有取值为$v_1, v_2, \ldots, v_n$，
    - 则依据这些取值将T划分为n个子集$T_1, T_2, \ldots, T_n$，
    - 建T的n个孩子结点T_i，并分别以X= v_i作为从T到T_I的分支标号。
  - 4）对每个T_i，以递归的方式建一棵以T_i为根的子树。
- 什么是信息增益？如何计算各属性的信息增益？

（1）用信息熵度量样例的纯度

$Entropy(S) = - p^{+}\log_2 p^{+} - p\text{-}\log_2 p^{-}$

S：样例的集合

p+：样例集中正例的比例

p-：样例集中反例的比例

例如：

Entropy([9+, 5-])

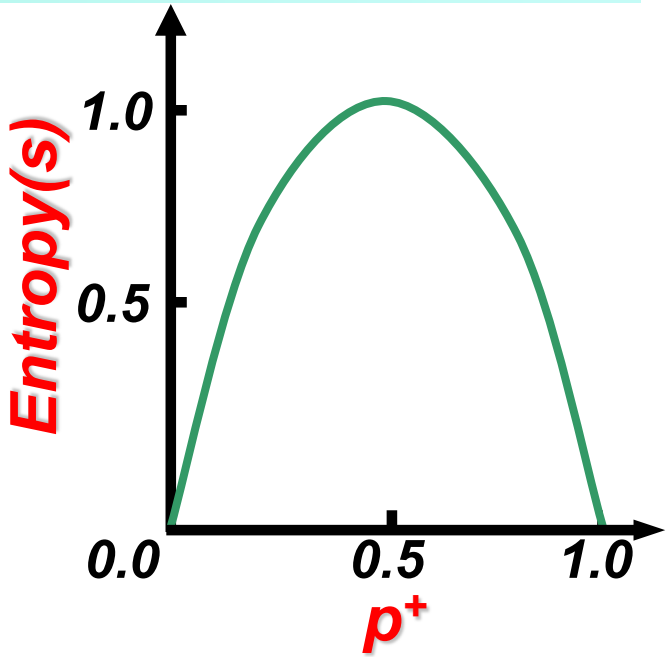
$= - (9/14)\log_2 (9/14) - (5/14)\log_2 (5/14)$

$= 0.940$

熵用于衡量类分布的倾斜度/不同节点中的不纯度

越纯则类分布就越倾斜

Entropy([9+, 5-])<*Entropy*([7+, 7-])

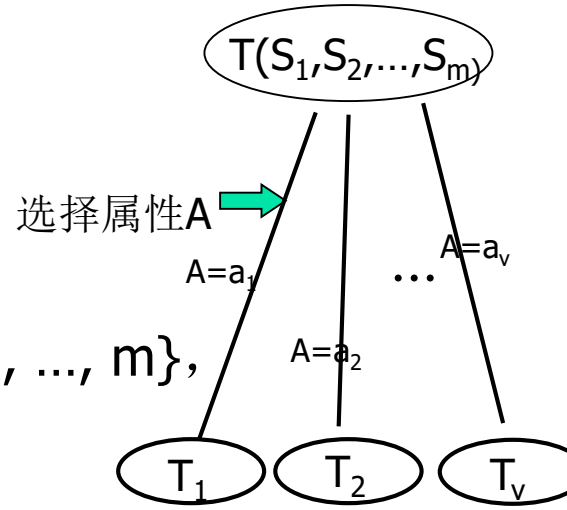


NO.	age	income	student	credit_rating	buys_computer
1.	<=30	high	no	fair	no
2.	<=30	high	no	excellent	no
3.	31…40	high	no	fair	yes
4.	>40	medium	no	fair	yes
5.	>40	low	yes	fair	yes
6.	>40	low	yes	excellent	no
7.	31…40	low	yes	excellent	yes
8.	<=30	medium	no	fair	no
9.	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no

（2）信息增益最大属性的选择

- 设集合T的元素个数为S，其中分别有s_i个元素属于C_i类，则集合T的信息熵为

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$



- 如果属性A的值域为 $\{a_1,a_2,\ldots,a_v\}$，
  - 则选择属性A分解T为v个子结点$T_1,T_2,\ldots,T_v$，
  - 其中每个T_i中分别有s_{ij}个实例属于C_j类，j= {1, …, m}，
  - 则选择属性A的期望信息熵(加权）为

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j}+\ldots+s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

- 则属性 A的信息增益为

$$Gain(A) = I(s_1,s_2,\ldots,s_m) - E(A)$$

- 属性 A的信息增益为节点划分前的熵与划分后的熵的差值

例：信息增益计算示例

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- 计算属性 *age的信息增益*:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

- I(2,3)表示 "age <=30" 在14个实例中占5个, 其中2 个的类属性为yes, 3 个为no. 因此 $Gain(age) = I(p,n) - E(age) = 0.246$
- 类似的有：Gain(income)=0.029, Gain(student)=0.151, Gain(credit_rating)=0.048
- 对每个属性来说，I(p,n)相同，仅需要比较E，为什么计算Gain

单选题 10分

设置

选出信息增益最大的属性

A 天况

B 温度

C 湿度

D 风况

编号	天况	温度	湿度	风况	分类
1	晴	热	大	无	N
2	晴	热	大	有	N
4	雨	中	大	无	Y
5	雨	冷	正常	无	Y
8	晴	中	大	无	N
10	雨	中	正常	无	Y

提交

2022年4月5日星期二

主观题 10分

设置

1）构造决策树
（假设属性选择顺序为：
天况—温度—湿度—风况）

2）对未知标签进行预测

编号	天况	温度	湿度	风况	分类
1	晴	热	大	无	N
2	晴	热	大	有	N
3	多云	热	大	无	Y
4	雨	中	大	无	Y
5	雨	冷	正常	无	Y
6	雨	冷	正常	有	N
7	多云	冷	正常	有	Y
8	晴	中	大	无	N
9	晴	冷	正常	无	Y
10	雨	中	正常	无	Y
11	晴	冷	正常	有	?
12	多云	中	大	有	?
13	多云	热	正常	无	?
14	雨	中	大	有	?

正常使用主观题需2.0以上版本雨课堂

作答

2022年4月5日星期二

4.3 决策树模型的评估

- 构造的决策树需要用新的数据（测试数据）来检验，

  ----针对正确分类率、误分类率、时间性能等指标，以此作为衡量求解结果的指标，以及作为修正的依据

- 训练集合与测试集合的划分 ----典型的划分有：

  2：1

  5：1

  9：1

- K-交叉验证（cross-validation）

分类应用中出现的问题——DT为例

- 实际应用中可能出现的问题：
- 从结果上看：
  - 不能运行
  - 精度不高
  - 误差大
  - 不稳定
- 原因：
  - 数据原因（缺失，噪音，非离散，取值过多，非相关）
  - 模型原因（模型优化）
  - 外部优化

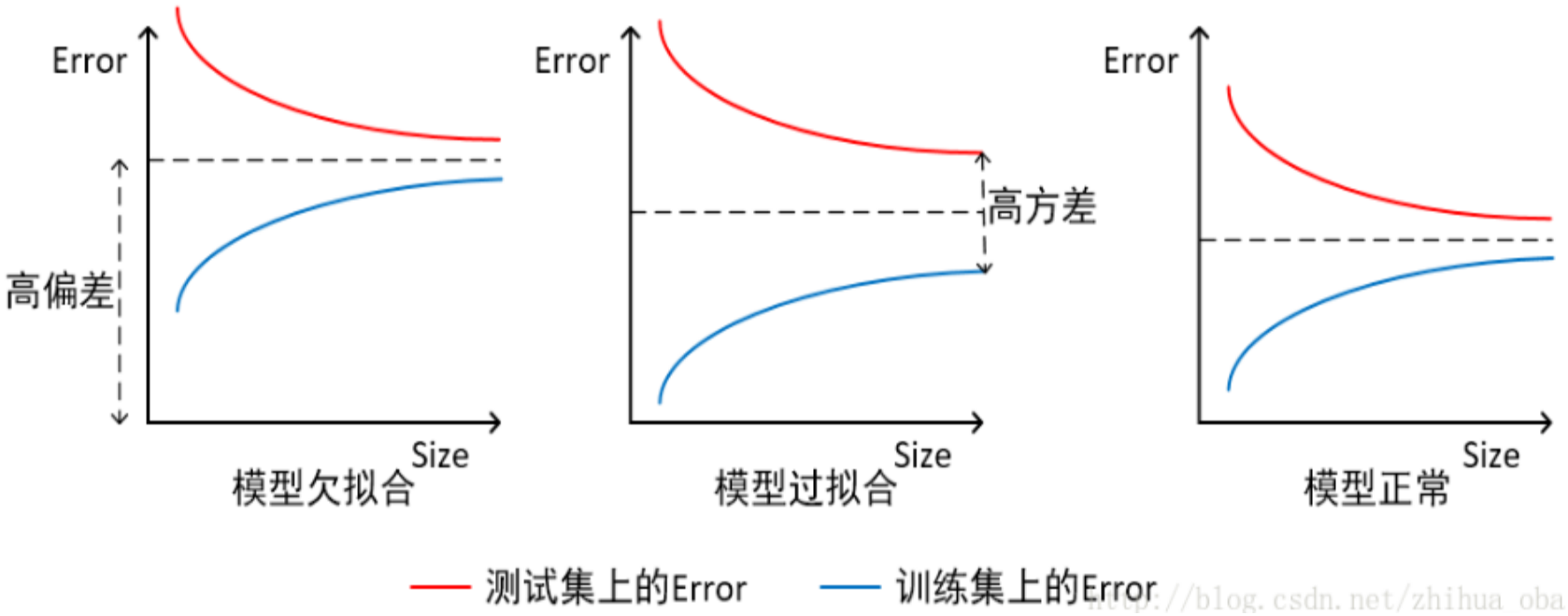
（5）基于决策树算法的进一步工作

- 允许处理连续型属性
  - 通过有效的离散化方法，
    将连续型属性转化为新的离散型属性，再做进一步处理
- 处理缺失属性值
  - 使用出现最频繁的属性值代替
  - 使用每一个属性的可能出现的取值来代替
- 决策树的优化：
  - 优化的决策树具有：
    - 正确分类率高，误分类率低，
    - 抗噪性能好，
    - 结构合理，
    - 规则长度小等特点

优化策略：
——剪枝；
——优化节点；
——属性选择策略；

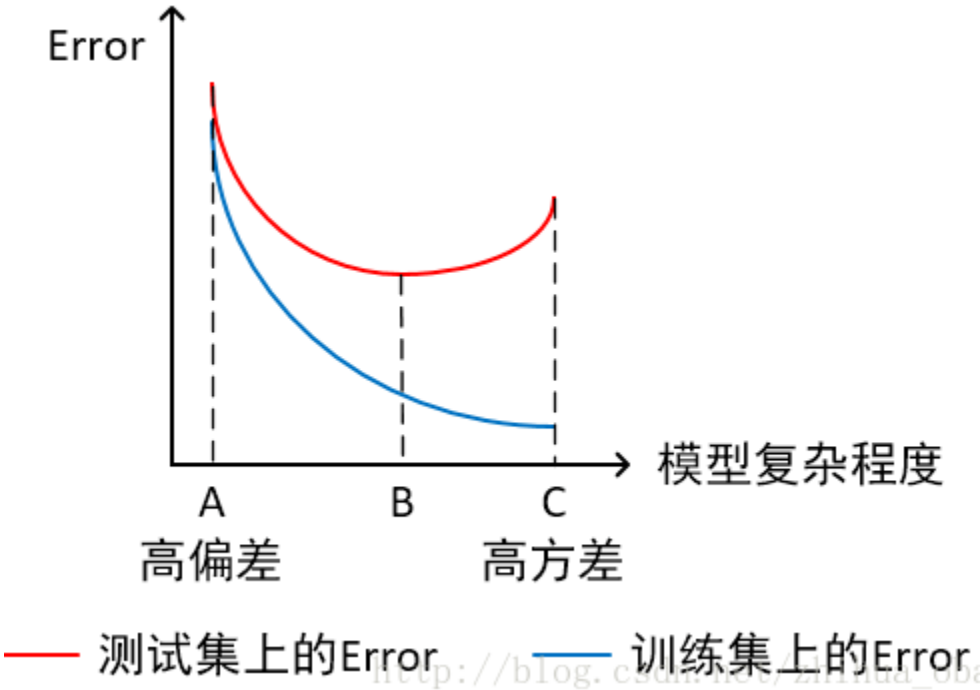
4.3 决策树模型的评估

- 模型的误差：具有低训练误差和低泛化误差
- 欠拟合：训练误差高，泛化误差高
- 过拟合：训练误差低，泛化误差高



4.3 决策树模型的评估

- 过拟合：训练误差低，泛化误差高
- 原因：
  - 噪音
  - 不相关数据
  - 样本代表性不足



大量的候选属性和少量的训练记录导致了过拟合

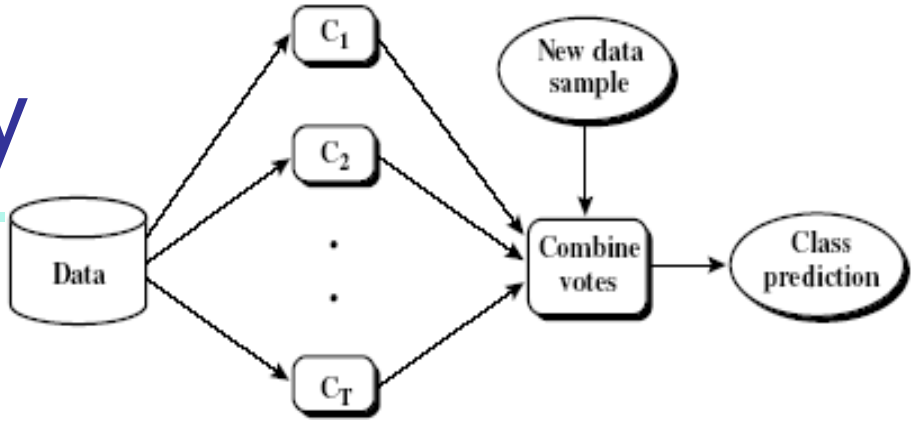
4.3 How to avoid overfitting—数据角度

- 估计泛化误差
  - Tranining error≈generalization error
  - K-cross validation
- 结合模型的复杂度
  - 节点个数
  - 最小描述长度
- 其他方法
  - 重新清洗数据
  - 增加训练样本数量
  - early stoping/减少迭代次数
  - 添加噪声数据。

4.3 如何处理决策树的过拟合—模型优化

- 过拟合(Overfitting)： 一棵建成的决策树可能对训练集合过度匹配
  - 过多的分支, 其中一些可能反映的是由噪音数据或者错误而导致的异常
  - 对未知数据的预测精度低下
- 避免过拟合的两种方法
  - 预剪枝: 在早期就停止树的生长—如果发现进一步分裂会导致某标准低于给定的阈值，则停止此次分裂
    - 难点是如何确定此阈值
  - 后剪枝:从一棵 “完全生长” 的树中删除一些子树，得到一系列不完全树
    - 使用与训练数据不同的数据集来判断哪一个才是 “最优剪枝树”

4.4 Ensemble Methods: Increasing the Accuracy

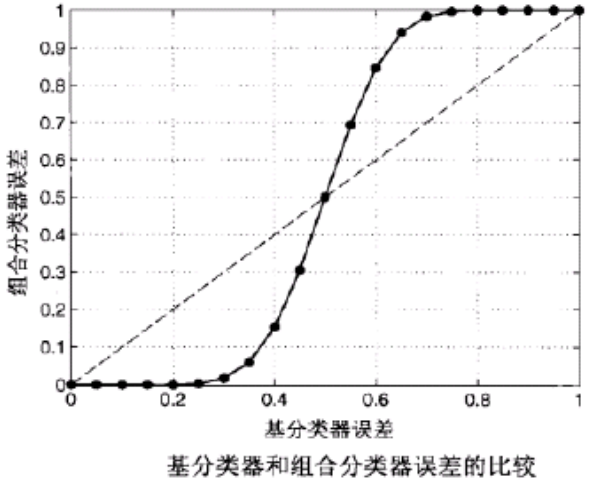


- Ensemble methods
  - 组合多个模型形成一个新模型以获得更高的分类精度
  - Why?

两个条件：

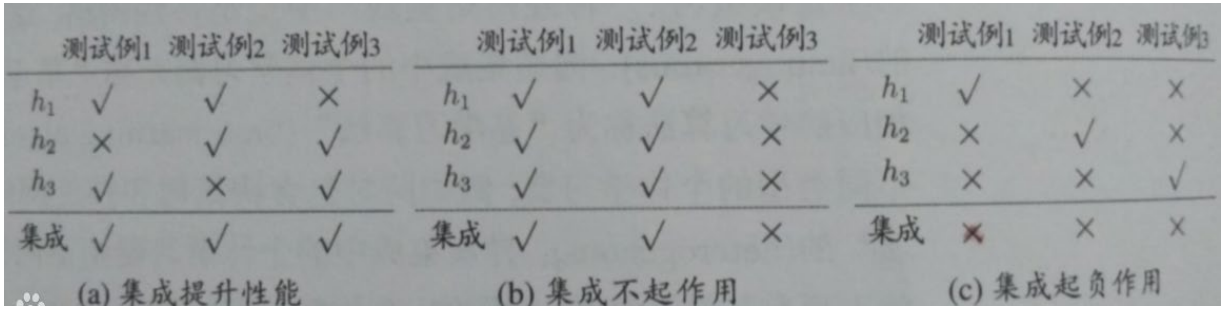
1）基分类器具有差异性/相互独立

2）基分类器性能好于随机猜测分类器



基分类器和组合分类器误差的比较

适用于性能不稳定的分类器



	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集成	√	√	√

(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

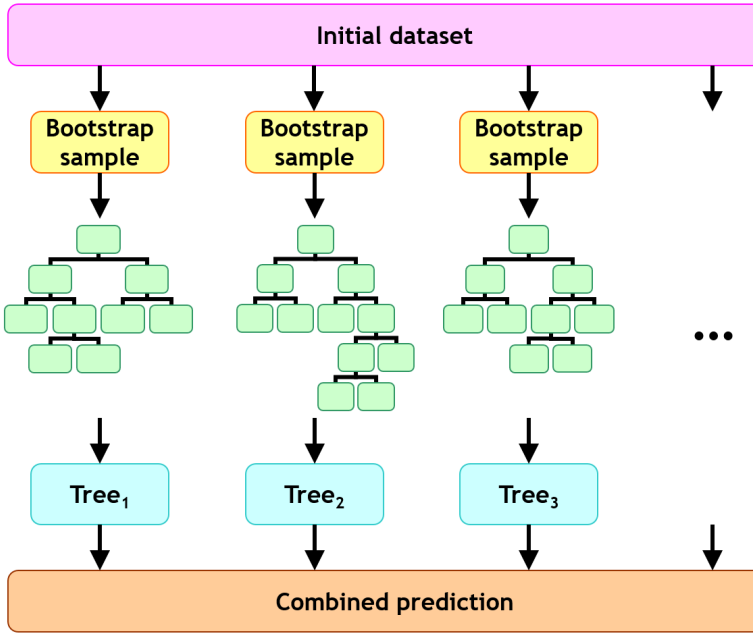
(c) 集成起负作用

4.5 Random Forest

- RT两个步骤：
- 封包(Bagging)
  - Bagging又称为bootstrap aggregation
  - Bootstrap抽样：设训练集中有N个样本，有放回的从中随机抽取N′≤N次组成新的训练集
  - Bootstrap aggregation：进行k次Bootstrap抽样得到k个独立的训练集合，在每个新训练集上独立训练分类器，分类时统计k个分类器的投票，并将“最多的”投票作为分类结果
- 随机向量
  - 在决策树的结点分割时，从所有属性中随机选择m个属性，再从这m个属性中确定最佳分割属性—半随机森林
  - m个属性随机选择分割属性——完全随机森林

4.5 Random Forest

- 构造算法
  - 假设要构建k个决策树，对每一个决策树执行以下操作：
    - 通过bootstrap抽样组成一个新的训练集
    - 在这个新训练集上构建一个(多个)无剪枝的决策树
    - 在决策树的结点分割时，从所有属性中随机选择m个属性，再从这m个属性中确定最佳分割点
  - 分类时统计k个决策树的分类结果，并将“最多的”类别作为分类结果
- 参数选择
  - k值：构建决策树直到错误率不再降低
  - m值：建议值为$\sqrt{M}$(M为所有属性个数)，然后以一半和2倍的方式枚举
    - k过大，相关性增加
    - F集增大，则相关性增加



4.5 Random Forest

- 优点
  - 对过拟合鲁棒
  - 对噪音鲁棒
  - 能够处理缺失属性值
  - 训练速度快

- 缺点
  - 特征选择过程不明显
  - 在较小的训练数据集上性能较差

Ensemble 小结

- 原理：多个不同的分类器合作效果好于单一分类器
- 原则：
  - 分类器间的差异性
  - 基模型性能好于随机
- 如何产生差异性
  - 数据差异：bagging或boosting——随机抽样，放回抽样，迭代概率抽样；RF
  - 模型差异：同一数据不同分类器
- 合并过程：
  - Vote
  - Weighted vote